P.B.5818 - Patentlaan 2
2280 HV Rijswijk (ZH)
☎ +31 70 340 2040
TX 31651 epo nl
FAX +31 70 340 3016

Europäisches Patentamt | **European Patent Office** | **Office européen des brevets**

Generaldirektion 1 | Directorate General 1 | Direction Générale 1

Etorre, Yves Nicolas
Compagnie IBM France,
Département Propriété Intellectuelle
06610 La Gaude
FR

IBM
FRANCE INTELLECTUAL PROPERTY DEPT
19 SEP. 2000
MD
ACTION RJG

Datum/Date
14/09/00

J1046 U.S. PTO
09/925353
08/09/01

Zeichen/Ref./Réf.	Anmeldung Nr./Application No./Demande n°./Patent Nr./Patent No./Brevet n°.
FR9-1999-0105RJ	00480074.4 2201
Anmelder/Applicant/Demandeur/Patentinhaber/Proprietor/Titulaire	
International Business Machines Corporation	

Übersendung von/Transmission of/Envoi de Antrag vom/Request dated/Requête du 30/08/00

- [ ] Kopien bei Akteneinsicht nach Regel 94(3) EPÜ
  Copies in the case of inspection of files pursuant to Rule 94(3) EPC
  Copies en cas d'inspection publique selon la règle 94(3) CBE
- [ ] Beglaubigung
  Certification
  Certification
- [X] 1 Prioritätsbeleg(e)/priority document(s)/document(s) de priorité R. 94(4)
- [ ] ___ Ausfertigung(en) der Patenturkunde nach Regel 54(2) EPÜ
  Duplicate of the patent certificate pursuant to Rule 54(2) EPC
  Duplicata du certificat de brevet, selon la Règle 54(2) CBE
- [ ] Auszug aus dem Register nach Regel 92(3) EPÜ
  Extract from the register pursuant to Rule 92(3) EPC
  Extrait du registre selon la Règle 92(3) CBE
- [ ] Auskunft aus den Akten nach Regel 95 EPÜ
  Communication of information contained in the files pursuant to Rule 95 EPC
  Communication d'informations contenues dans la dossier selon la Règle 95 CBE
- [ ] Akteneinsicht nach Regel 94(2) EPÜ
  Inspection of files pursuant to Rule 94(2) EPC
  Inspection publique selon la Règle 94(2) CBE

HOORN F J (TEL:3581)

09/925353

#4

Europäisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

00480074.4

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

Hatten

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 14/09/00
LA HAYE, LE

EPA/EPO/OEB Form 1014 - 02.91

Europäisches Patentamt | European Patent Office | Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 00480074.4
Demande n°:

Anmeldetag:
Date of filing: 09/08/00
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armonk, N.Y. 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
System and method for enhancing load controlling in a clustered web site

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

SYSTEM AND METHOD FOR ENHANCING LOAD CONTROLLING IN A CLUSTERED WEB SITE

Field of the Invention

The present invention relates generally to the global Internet network and more particularly to those of the Internet servers of World Wide Web (WWW) sites organized as a cluster or group of servers forming a single entity.

Background of the Invention

The Internet has grown so rapidly over the past years that many companies, utilizing Web sites for their daily activities and as an efficient and attractive interface with their

customers or users, are facing the problem of handling their own share of the explosive overall Internet traffic growth. If this growth is however not properly handled, users get slow response or refused connections, creating an unsatisfactory user experience which may be detrimental to the company owning such a Web site and may have catastrophic commercial impacts. Moreover, under critical load conditions, when too many requests for connection are received and/or because the service of the current requests is too demanding for the site available computing resources this latter can become unstable or even just collapse. In order to prevent these problems to occur Web sites have soon been organized under the form of clusters of servers so scalability and full availability can be warranted through redundancy. Individual servers forming a cluster can thus be selectively replaced e.g., in case of failure and/or their number increased to cope, as needed, with a growing demand. This, being accomplished without requiring any service interruption from the Web site since, only the affected individual servers need to be shut down while the others are still up and running keep serving users' requests. Load-balancing i.e., the ability of spreading in a fair manner the overall workload over the active individual servers, was the key component to permit this to effectively happen. As an example, among other products commercially available, ND (Network Dispatcher), the load balancing component of WebSphere Performance Pack by IBM (International Business Machines Corp., Armonk, N.Y., the USA) has been developed to address these limitations and provide customers with advanced functions to meet their site's needs. A description of this particular product and functions it provides can be found e.g., in 'Load Balancing with IBM SecureWay Network Dispatcher', published by ITSO (International Technical Support Organization) of IBM Corporation, Research Triangle Park, North Carolina, 27709, the USA, under reference SG24-5858. If the early versions of Web traffic load balancers were simply performing round-robin, rotating the resolution of individual server names among a

hard-coded list of IP addresses for Web application servers, balancing was hardly optimal since all user requests were considered as equal and, the actual availability of the individual servers and the workload on them was not taken into account; the following issues of load balancing software included a more intelligent round-robin method along with a user-specified approach to distributing TCP/IP session requests. Thus, newer load balancing products, like ND here above mentioned, improve the performance of servers by basing their load balancing decision not only on the servers' availability, capability and workload, but also on many other new user-defined criteria as well. As a result of this greater flexibility, Web sites can now take advantage of differentiated qualities of service, based on request origin, request content and overall load on the system while the entire load balancing operation is transparent to end users and other applications. This is very useful for applications such as e-mail servers, World Wide Web (WWW) servers, distributed parallel database queries, and many other TCP/IP applications. When used with Web servers, it can help maximize the potential of a Web site by providing a powerful, flexible, and scalable solution to peak-demand problems. Then, ND and the more advanced load balancer products are able to balance the load on clusters of servers, servers within a local area network (LAN) or even over a wide area network (WAN) using a number of weights and measurements that are dynamically set by a dispatcher component which provides load balancing at the level of specific services, such as HTTP (Hyper Text Transport Protocol), FTP (File Transfer Protocol), SSL (Secure Sockets Layer) and Telnet (the virtual terminal protocol based on TCP/IP). Load balancing on servers within a local or wide area network can be performed using a DNS (Domain Name System) round-robin approach or a more advanced user-specified approach while client Web requests can also be directed to specified servers by comparing the content of the request to a predefined set of rules.

However, whichever level of sophistication the newer load-balancing functions have reached they all still fail handling an important aspect of load balancing namely, the lack of provision to allow applications running on the individual servers or on users machines to pass feedback information to the load-balancing function so that this latter may react accordingly. A typical example of this is pacing. If an application, although running on a server which is far to be over utilized however, cannot handle, by design or because of any other consideration, more than a given number of requests over a certain period of time to work properly load-balancing should better be informed when this is reached so that it may pace the rate at which requests are forwarded to this particular application even though server resources are far to be fully exploited.

Object of the Invention

Thus, it is a broad object of the invention to permit that, for the fastest growing and most widely used of the IP application i.e., the Web and HTTP, application scheduling and control facilities become an integral part of load-balancing functions so as to achieve a better utilization of all the resources available at sites where multiple servers cooperate to deliver this service.

Further objects, features and advantages of the present invention will become apparent to the ones skilled in the art upon examination of the following description in reference to the accompanying drawings. It is intended that any additional advantages be incorporated herein.

Summary of the Invention

A system and a method for enhancing load controlling of a Web site are disclosed. Web site is assumed to be comprised of a plurality of individual servers also, including a Network Control Scheduler (NCS). The Web site uses the Hyper Text Transport Protocol (HTTP). Then, invention allows that any server out of the plurality of individual servers may issue instructions to NCS to which this latter has to comply with. The instructions are passed to NCS in a NCS-control HTTP header including directives to be obeyed by NCS. Directives falls in three categories namely: flow-control directives, sharing directives and NCS-queuing directives. They optionally include a filter which limits their scope of application.

Hence, the invention permits that Web sites using HTTP and where multiple servers cooperate thus, forming a cluster of servers, application scheduling and control facilities become an integral part of load-balancing functions so as to achieve a better utilization of all site resources.

Brief Description of the Drawings

Figure 1 illustrates prior art.

Figure 2 introduces the Network Control Scheduler (NCS) per the invention as part of a cluster servers.

Figure 3 describes the directives to which NCS must obey.

Figure 4 shows a scenario involving the share directive.

Figure 5 shows a scenario involving the NCS-queuing directive (lock and unlock).

Detailed Description of the Preferred Embodiment

Figure 1 illustrates a typical situation of the prior art where the invention better applies showing a Web server [100] implemented under the form of a cluster of individual servers [101], [102] and [103] fed from a load-balancer [120] aimed at spreading the load generated from remote clients e.g., [130] issuing requests through a private and/or a public IP network [110] or any combination of. When an initial request [131] reaches the server [100] load balancer [120] must decide, on its own, based on what it knows of the current load over the individual servers, to have the initial request [131] served on an individual server e.g., [103]. Thus, a session [135] is open between user [130] and the individual server [103]. However, no further monitoring of subsequent user requests and responses, exchanged during the session, are susceptible to be used to alter the exchange of information so as to dynamically better adapt to the actual behavior of either the particular user [130], the individual server [103] or the cluster of servers as a whole while it would be highly desirable that the exchanges of data during a session such as [135] be monitor able and adjustable at application level through the exchange of scheduling and pacing commands in order to obtain a better regulation of the flows resulting in a better utilization of the network and server resources.

Figure 2 HTTP (Hyper Text Transport Protocol) is the primary protocol used for transferring Web documents. It is a request/response protocol i.e., a client [200] sends a request [205] which ends up in a server e.g., [210], and the server sends back a response [215]. There are no multiple-step handshakes in the beginning as with some other protocols. Server is shown to be here comprised of four individual servers [210, 212, 214, 216] possibly equipped with a front-end load balancing [222] of the kind discussed in the background section and in figure 1 however, also including a

NCS (Network Control Scheduler) function [224] per the invention. An HTTP request [230], forwarded [240] to the server [210] through NCS function [224], consists of a method e.g., the GET method [231], a target URL (Uniform Resource Locator) [232], a protocol version identifier [233], and a set of headers further discussed in the following. The method specifies the type of operation. The most common method, GET [231], is used to retrieve documents. Headers contain additional information to the request and responses.

Much more on this can be found in the version 1.0 of HTTP protocol which is documented in the Informational RFC (Request For Comment) #1945 titled "Hypertext Transfer Protocol-HTTP/1.0." of the IETF (Internet Engineering Task Force). HTTP in general is also discussed in numerous other publications such as in a book by Ari LUOTONEN, titled "Web Proxy Servers", published by PRENTICE HALL, ISBN 0-13680612-0.

An HTTP response [250] consists of a protocol version identifier [251], a status code (200) [252], a human-readable response status (OK) [253] and response headers [254] followed by the requested resource content (not shown) since request was assumed to be a GET in this example. Headers are used to include additional information to both requests and responses. The version 1.1 of HTTP specification defines 46 standard headers. An example of a general header (a general header can be found either in a request or in a response while there are specific request or response headers), is for example the "Cache-Control" header which is used to control various aspects of caching a key function to be supported in many proxies. In addition to the standard headers new applications are allowed to have their own specific headers so as HTTP can easily be extended. Therefore, the invention assumes that a set of HTTP headers such as [254], aimed at enabling flow-control between clients and Web servers, are specified. The new HTTP header shown in this particular example is thus aimed at informing NCS [224] to have to increase its rate of requests when they are directed to the individual server named

'Hercules'. Hence, this server becomes able to dynamically inform those using its resources that it can indeed, for the time being, processes more requests than presently ask to do, this information may be used by NCS and may not need to be included however in the chained response [260] to the end-client. It is worth noting here that the syntax of the new HTTP headers of this description of the invention follows generally what is used in RFC (Request For Comment) 2616 of the IETF (Internet Engineering Task Force) specifying the version 1.1 of HTTP. Especially, RFC 2616 makes use of a so-called BNF (Backus-Naur) form of notation which is occasionally utilized in the following too.

Figure 3 shows the defined directives of a new HTTP header here after referred to as "NCS-Control" [300] header that must be obeyed by NCS component. The overall intent of these directives is to extend the scope of the HTTP protocol replies in order they can convey flow control information, as well as cluster level control and communication commands, to a NCS function implemented as a front end to a cluster of servers. The new NCS-Control HTTP header is comprised, in the general case, of a directive [310] and a filter [320], that limits the range of application of the directive, plus additional data when necessary. The filter operates on HTTP objects, such as the origin server, known by its DNS (Domain Name System) name or IP address, the client, the HTTP headers, the cookies (described in RFC 2109), and the URL (Uniform Resource Locator). The filter syntax [330] includes a type, an object name and a string in the form of a regular expression that is to be matched with the occurrence of the HTTP object in the request. The default is 'all', meaning that the directive applies to all further requests. There are three types of filters i.e., URL, Cookie and Headers, depending on the object it operates on. Well know object names are added to represent the URL content: the locally significant part (URI), the web server (Host) and the client (Client).

A first group [301] of NCS-Control header directives deals with flow control. NCS function is then made capable of controlling the rate at which requests are passed to the cluster of servers and expressed as a number of requests per unit of time e.g., per second. Also, the overall number of requests that are being served at any moment can be controlled by specifying a window setting a number of requests that are allowed to be processed simultaneously. Hence, invention adds directives to increase and decrease both window and rate. Thus, directives are: "increase-rate" , "decrease-rate", "increase-window", and "decrease-window". An example of this, also shown in figure 2 [254], is thus:

NCS-Control: increase-rate; URL=(Host, Hercules)

which causes NCS to allow more connections per second globally on the member of the cluster whose name is 'Hercules'.

A second group of directives [302] enables the sharing of information among all members within a cluster of servers so as they can all become aware of a situation or event regarding a member. To achieve this, HTTP is used as a communication means and NCS function acts as a central communication device. Then, any member in the cluster is allowed to deposit an HTTP header in NCS. This latter is added to all subsequent requests, issued from NCS and matching a provided filter, till HTTP header is cleared. The directives used to achieve this are "share" and "clear" of which an example is:

NCS-Control: share=(Username, Pascal); Cookie=(Session, 9098098)

Thus, "share" causes NCS to add an HTTP header e.g., 'Username, Pascal' in any subsequent request containing a cookie named 'Session' and having a value of '9098098' while "clear" does the opposite i.e., remove it.

A third group of directives [303] is devoted to queue management within NCS. Two directives of this category are "lock" and "unlock". They are aimed at allowing NCS function to be able to control whether the service of a cluster resource, identified with a filter, can be performed

immediately or should rather be delayed so that they are temporarily locked. An example of it being:

NCS-Control: lock; URL=(URI, "/appli/*")

which causes NCS to lock all the resources of a particular application e.g., "/appli/" and where URI is the locally significant part of the URL. Either a time out or an unlock command can clear the lock.

Figure 4 further illustrates the second group of directives, which deals with the sharing of information among the cluster members. As an example, an existing user session is identified with a cookie named 'Session' and having a value of '9098098'. Then, issuing following command:

NCS-Control: share=(Username, Pascal); Cookie=(session, 9098098)

Hercules instructs NCS [400] to add an HTTP header named 'Username' with a value of 'Pascal' to any subsequent request if a cookie named 'session' is present with a value of '9098098' [410]. In which case, an application running on Mercury [420], or on any one of the individual servers of the cluster, can directly extract the user name from the HTTP flow without having to go to the session to do so. Later (not shown), upon issuing:

NCS-Control: clear=Username; Cookie=(session, 9098098)

Hercules commands NCS to undo the previous command with the same scope, at the time the session is removed by the application server.

Figure 5 illustrates the third group of directives, which involves queue management within NCS. Upon receiving a request from a client e.g., Client_1 [500] to control URL, Hercules instructs NCS [510] to delay any further request to the WEB application path '/appli/', issuing:

NCS-Control: lock; URL=(URI, "/appli/*")

Hence, when a request from e.g., Client_3 [520] arrives later on it is queued in NCS. When a condition is reached, Client_1 comes back to the control application and, as a result, Hercules may release the lock [530] by issuing:

NCS-Control: unlock; URL = (URI, "/appli/*")

At that point, the request from Client_3 is passed to the cluster [540], and may be served by Mercury. This mechanism allows to define e.g., fall back URLs in NCS so that when combined with the "lock" directive, after a given period of time has elapsed, the queued requests would be actually passed to a fall back service.

Claims:

What is claimed is:

1. A method for enhancing load controlling of a Web site [220] including a plurality of individual servers [210, 212, 214, 216] and a Network Control Scheduler (NCS) [224], said Web site using the Hyper Text Transport Protocol (HTTP), said method comprising the steps of:

in any one server out of said plurality of individual servers:

issuing instructions to said NCS [250];

in said NCS [224]:

receiving said instructions from said any one server;

complying with received said instructions.

2. The method of claim 1 wherein the step of issuing instructions includes the step of:

passing said instructions to said NCS in a NCS-control HTTP header [300], said passing step including the further step of:

including directives [310] to be obeyed by said NCS;

optionally including a filter [320] to limit the scope of application of said directive.

3. The method according to any one of the previous claims wherein said directives includes:

flow-control directives [301];

sharing directives [302];

NCS-queuing directives [303].

4. The method according to any one of the previous claims wherein said flow-control directives [301] include:

an increase-rate directive to require said NCS to increase the rate at which requests to said any one server are sent;

a decrease-rate directive to require said NCS to decrease the rate at which requests to said any one server are sent;

an increase-window directive to require said NCS to increase the number of jobs allowed to be simultaneously processed in said any one server;

a decrease-window directive to require said NCS to decrease the number of jobs allowed to be simultaneously processed in said any one server;

5. The method according to any one of the previous claims wherein said sharing directives [302] include:

a share directive aimed at enabling an information sharing within all members of said plurality of individual servers and said NCS;

a clear directive aimed at clearing a previous said information sharing.

6. The method according to any one of the previous claims wherein said NCS-queuing directives [303] include:

a lock directive aimed at locking resources identified by a said filter;

an unlock directive aimed at releasing previously locked said resources.

7. A system, in particular a Network Control Scheduler [224] comprising means adapted for carrying out the method according to any one of the previous claims.

8. A computer-like readable medium comprising instructions for carrying out the method according to any one of the claims 1 to 6.

THIS PAGE BLANK (USPTO)

SYSTEM AND METHOD FOR ENHANCING LOAD CONTROLLING IN A CLUSTERED WEB SITE

Abstract

The invention discloses a system and a method for enhancing load controlling of a Web site using HTTP and assumed to be comprised of a plurality of individual servers also, including a Network Control Scheduler (NCS). The invention allows that any individual server may issue instructions to NCS. They are passed to it in a NCS-control HTTP header including directives that must be obeyed. Directives falls in three categories namely: flow-control directives, sharing directives and NCS-queuing directives. They optionally include a filter which limits their scope of application.

Hence, the invention permits that in Web sites using HTTP and organized as a cluster of servers, application scheduling and control facilities become an integral part of load-balancing functions so as to achieve a better utilization of all site resources.

Figure 2.

FR 9 99 105
DIAS et al.
1/5

PRIOR ART

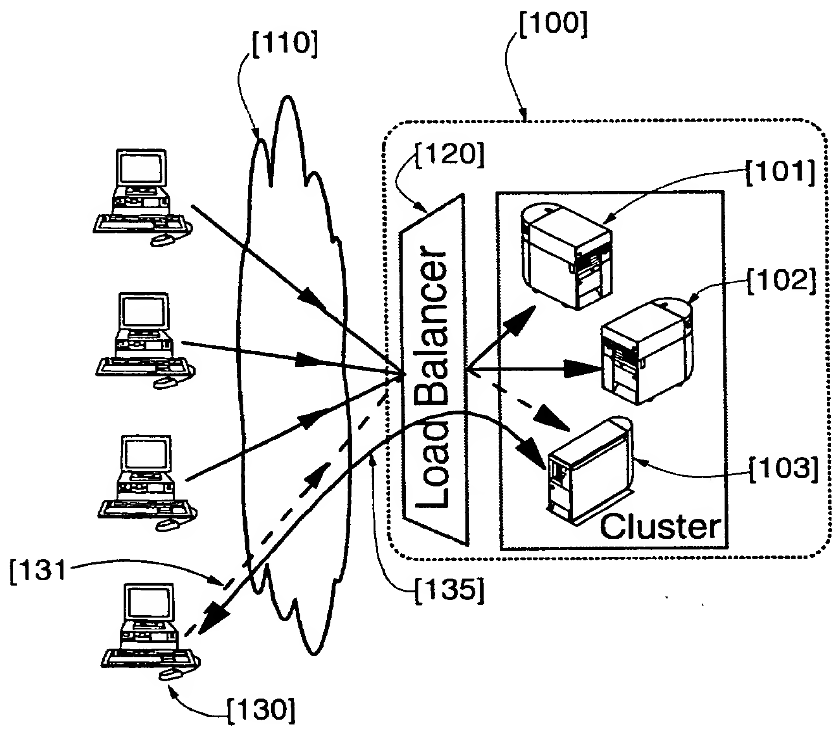

FIGURE 1

FR 9 99 105
DIAS et al.
2/5

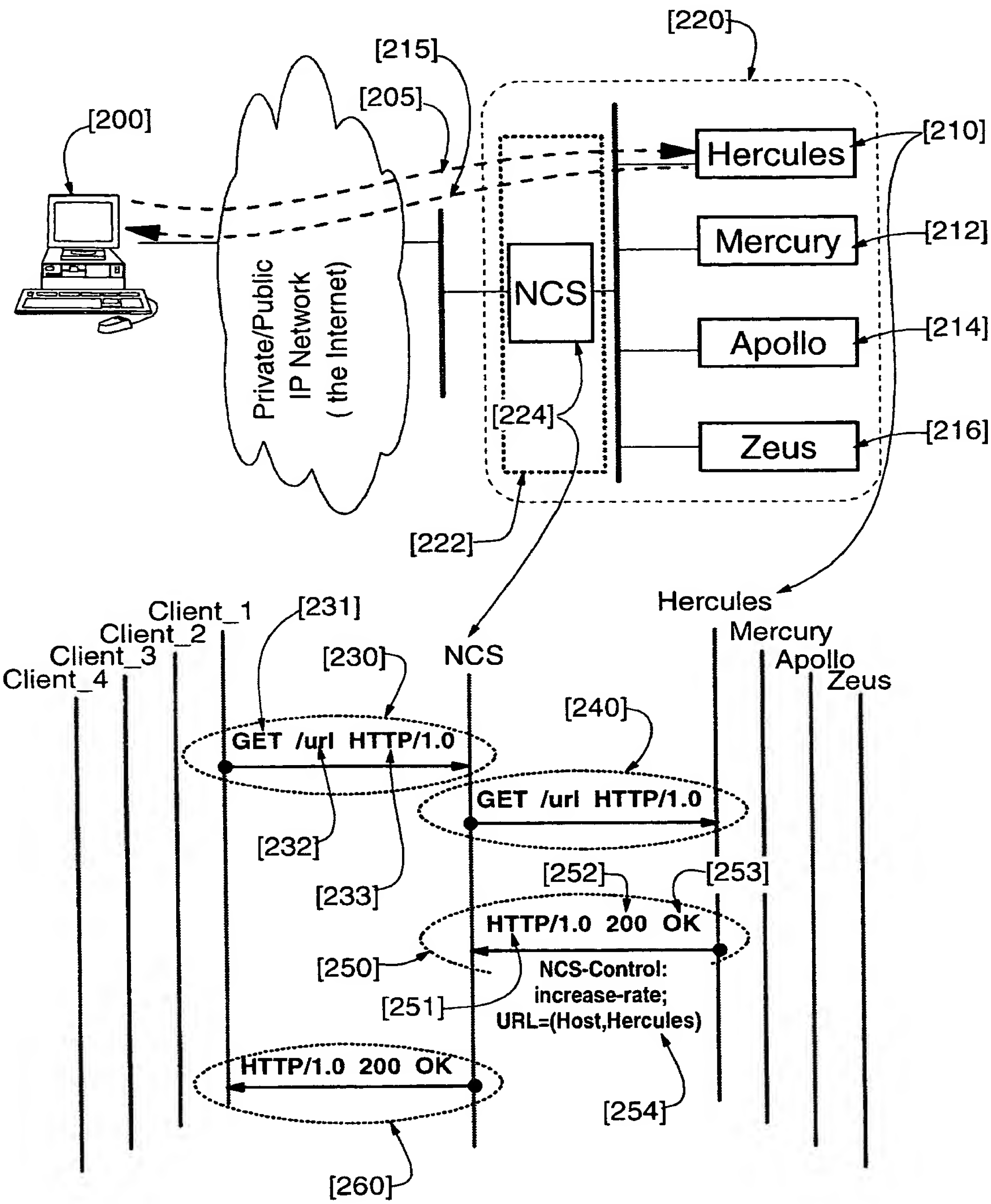

FIGURE 2

FR 9 99 105
DIAS et al.
3/5

[300]

	"NCS-Control" Header Directives	
	Increase-Rate:	causes NCS to allow more requests per unit of time over a member in a cluster of servers
	Decrease-Rate:	causes NCS to diminish the number of requests per unit of time over a member in a cluster of servers
[301]	Increase-Window:	causes NCS to allow a larger number of jobs to be handled simultaneously by a member in a cluster of servers
	Decrease-Window:	causes NCS to diminish the number of jobs handled simultaneously by a member in a cluster of servers
	Share:	causes NCS to enable sharing of information within the cluster of server
[302]	Clear:	causes NCS to clear a previous sharing of information
	Lock:	causes NCS to lock ressources according to a filter
[303]	Unlock:	causes NCS to release ressources previously locked

NCS-Control = "NCS-Control" ":" 1#(ncs-directive [filter])

[310] [320]

[330]

filter = ";" type ["=" (token | pair | quoted-string)]
type = ("URL" | "Host" | "Client" | "Cookie")
pair = "(" head "," val ")"
head = token
val = (token | quoted-string)

FIGURE 3

FR 9 99 105
DIAS et al.
4/5

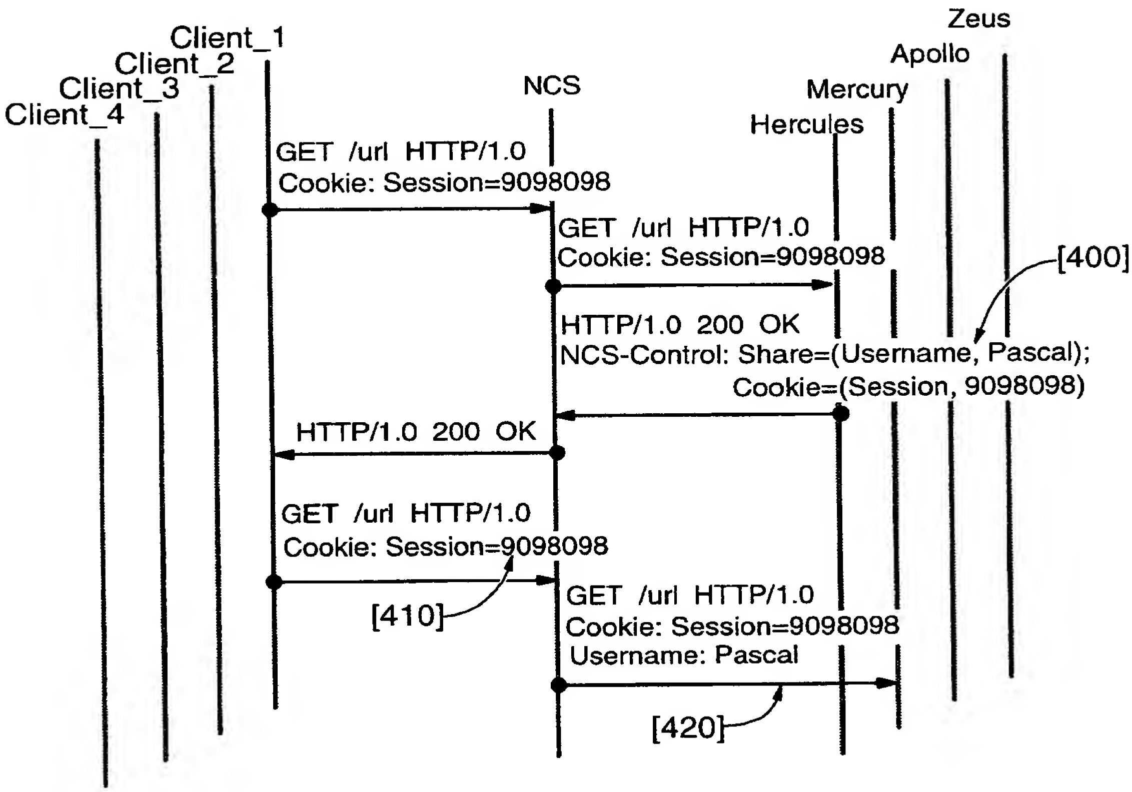

FIGURE 4

FR 9 99 105
DIAS et al.
5/5

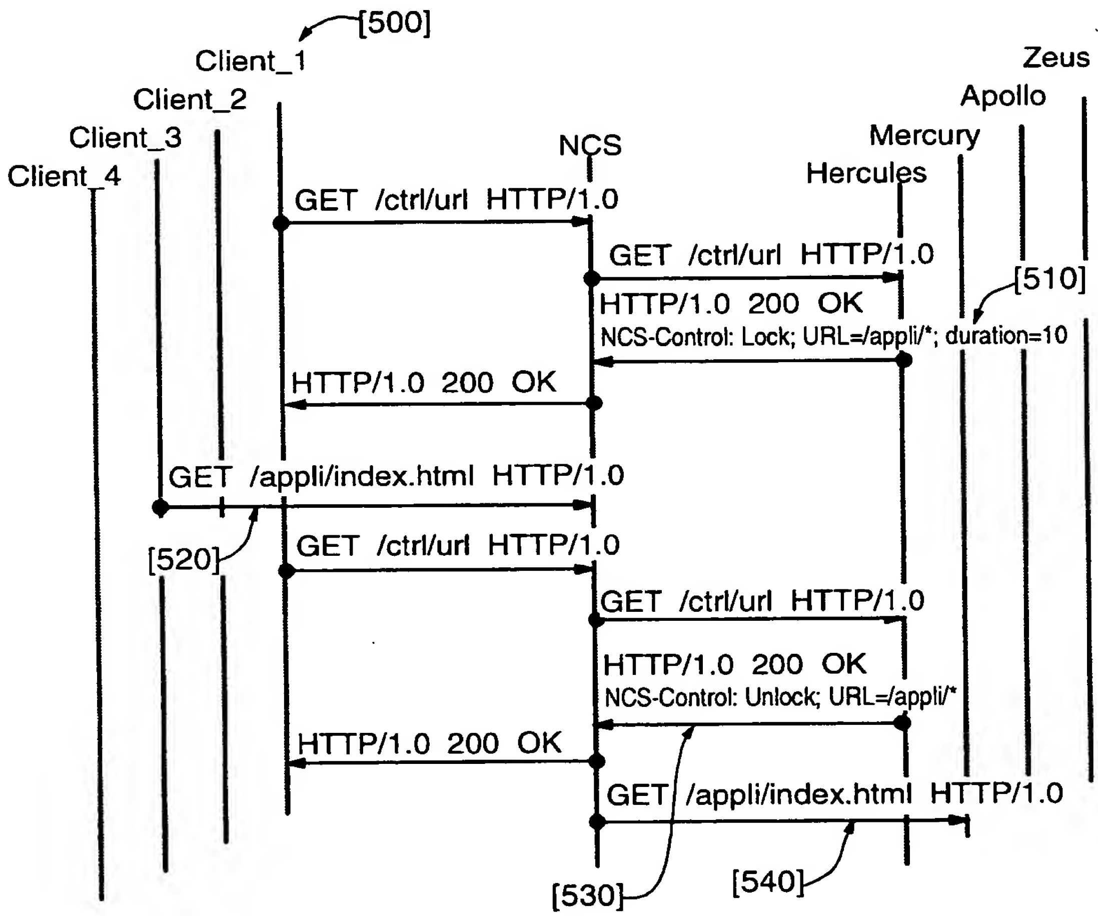

FIGURE 5